

**The Experience Factory:****Can it make you a 5?****or****What is its relationship to other Quality and Improvement Concepts?**

Victor R. Basili

Institute for Advanced Computer Studies  
Department of Computer Science  
University of Maryland

The concepts of quality improvement have permeated many businesses. It is clear that the nineties will be the quality era for software and there is a growing need to develop or adapt quality improvement approaches to the software business. Thus we must understand software as an artifact and software as a business.

Any successful business requires a combination of technical and managerial solutions. It requires that we understand the processes and products of the business, i.e., that we know the business. It requires that we define our business needs and the means to achieve them, i.e., we must define our process and product qualities. We need to define closed loop processes so that we can feedback information for project control. We need to evaluate every aspect of the business, so we must analyze our successes and failures. We must learn from our experiences, i.e., each project should provide information that allows us to do business better the next time. We must build competencies in our areas of business by packaging our successful experiences for reuse and then we must reuse our successful experiences or our competencies as the way we do business.

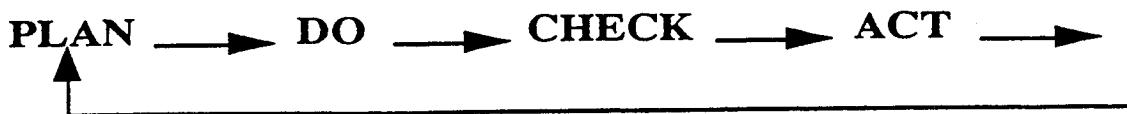
Since the business we are dealing with is software, we must understand the nature of software and software development. The software discipline is evolutionary and experimental; it is a laboratory science. Software is development not production. The technologies of the discipline are human based. There is a lack of models that allow us to reason about the process and the product. All software is not the same; process is a variable, goals are variable, etc. Packaged, reusable, experiences require additional resources in the form of organization, processes, people, etc.

There have been a variety of organizational frameworks proposed to improve quality for various businesses. The ones discussed here include:

**Plan-Do-Check-Act** is a quality improvement process based upon a feedback cycle for optimizing a single process model/production line. The **Experience Factory /Quality Improvement Paradigm** involves continuous improvement through the experimentation, packaging and reuse of experiences based upon a business's needs. **Total Quality Management** represents a management approach to long term success through customer satisfaction based on the participation of all members of an organization. The **SEI Capability Maturity Model** is a staged process improvement based upon assessment with regard to a set of key process areas until you reach a level 5 which represents a continuous process improvement. **Lean (Software) Development** represents a principle supporting the concentration of the production on "value added" activities and the elimination or reduction of "not value added" activities. In what follows, we will try to define these concepts in a little more detail to distinguish and compare them.

## Plan-Do-Check-Act Cycle (PDCA)

The approach is based upon work by W. A. Shewart [Sh31] and was made popular by W. E. Deming [De86]. The goal of this approach is to optimize and improve a single process model / production line. It uses such techniques as feedback loops and statistical quality control to experiment with methods for improvement and build predictive models of the product.



If a family of Processes ( $\mathcal{P}$ ) produces a family of Products ( $\mathcal{X}$ ) then the approach yields a series of versions of product  $\mathcal{X}$  (each meant to be an improvement of  $\mathcal{X}$ ), produced by a series of modifications (improvements) to the processes  $\mathcal{P}$ ,

$$\mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n \dots \rightarrow \mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n$$

where  $\mathcal{P}_i$  represents an improvement over  $\mathcal{P}_{i-1}$  and  $\mathcal{X}_i$  has better quality than  $\mathcal{X}_{i-1}$ .

The basic procedure involves four basic steps:

**Plan:** Develop a plan for effective improvement, e.g., quality measurement criteria are set up as targets and methods for achieving the quality criteria are established.

**Do:** The plan is carried out, preferably on a small scale, i.e., the product is produced by complying with development standards and quality guidelines.

**Check:** The effects of the plan are observed; at each stage of development, the product is checked against the individual quality criteria set up in the Plan phase.

**Act:** The results are studied to determine what was learned and what can be predicted, e.g., corrective action is taken based upon problem reports.

## Total Quality Management (TQM)

The term TQM was coined by the Naval Air Systems Command in 1985 to describe its Japanese style management approach to quality improvement [Fe90]. The goal of TQM is to generate institutional commitment to success through customer satisfaction. The approaches to achieving TQM vary greatly in practice so to provide some basis for comparison, we offer the approach being applied at Hughes. Hughes uses such techniques as Quality Function Deployment (QFD), design of experiments (DOE), and statistical process control (SPC), to improve the product through the process.

Identify needs --> ID Important items --> Make Improvements --> Hold Gains --> Provide Satisfaction



The approach has similar characteristics to the PDCA approach. If Process ( $P$ ) --> Product ( $X$ ) then the approach yields

$$P_0, P_1, P_2, \dots, P_n \longrightarrow X_0, X_1, X_2, \dots, X_n$$

where  $P_i$  represents an improvement over  $P_{i-1}$  and  $X_i$  provides better customer satisfaction than  $X_{i-1}$ .

### SEI Capability Maturity Model

The approach is based upon organizational and quality management maturity models developed by R Likert [Li67] and P. Crosby [Cr80], respectively. A software maturity model was developed by Ron Radice, et. al. [RaHaMuPh85] while he was at IBM. It was made popular by Watts Humphrey [Hu89] at the SEI. The goal of the approach is to achieve a level 5 maturity rating, i.e., continuous process improvement via defect prevention, technology innovation, and process change management.

As part of the approach, a 5 level process maturity model is defined. A maturity level is defined based on repeated assessment of an organization's capability in key process areas. Improvement is achieved by action plans for poorly assessed processes.

Level	Focus
5 Optimizing	Continuous Process Improvement
4 Managed	Product & Process Quality
3 Defined	Engineering Process
2 Repeatable	Project Management
1 Initial	Heros

Thus, if a Process ( $P$ ) is level  $i$  then modify the process based upon the the key processes of the model until the process model is at level  $i+1$ .

The SEI has developed a Process Improvement Cycle to support the movement through process levels. Basically it consists of the following activities:

Initialize

- Establish sponsorship
- Create vision and strategy
- Establish improvement structure

For each Maturity level:

- Characterize current practice in terms of key process areas
- Assessment recommendations
- Revise strategy (generate action plans and prioritize key process areas)

For each key process area:

- Establish process action teams
- Implement tactical plan, define processes, plan and execute pilot(s), plan and execute institutionalize

Document and analyze lessons  
Revise organizational approach

## Lean Software Development

The approach is based upon Lean Enterprise Management, a philosophy that has been used to improve factory output. Womack, et. al. [WoJoRo90], have written a book on the application of lean enterprises in the automotive industry. The goal is to build software using the minimal set of activities needed, eliminating non essential steps, i.e., tailoring the process to the product needs. The approach uses such concepts as technology management, human centered management, decentral organization, quality management, supplier and customer integration, and internationalization/regionalization.

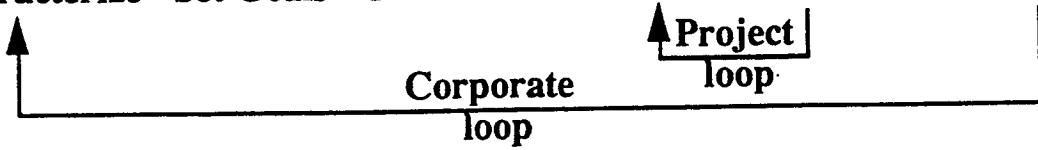
Given the characteristics for product  $\mathcal{V}$ , select the appropriate mix of sub-processes  $p_i, q_j, r_k, \dots$  to satisfy the goals for  $\mathcal{V}$ , yielding a minimal tailored process  $\mathcal{P}\mathcal{V}$  which is composed of  $p_i, q_j, r_k, \dots$

Process ( $\mathcal{P}\mathcal{V}$ ) -----> Product ( $\mathcal{V}$ )

## Quality Improvement Paradigm

This approach has evolved over 17 years based upon lessons learned in the SEL [Ba85a], [Ba89], [BaRo87], [BaRo88], [BaCaMc92]. Its goal is to build a continually improving organization based upon its evolving goals and an assessment of its status relative to those goals. The approach uses internal assessment against the organization's own goals and status (rather than process areas) and such techniques as GQM, model building, qualitative/quantitative analysis to improve the product through the process.

Characterize - Set Goals - Choose Process - Execute - Analyze - Package



If Processes ( $\mathcal{P}_X, Q_Y, R_Z, \dots$ ) ---> Products ( $X, Y, Z, \dots$ ) and we want to build  $\mathcal{V}$ , then based upon an understanding of the relationship between  $\mathcal{P}_X, Q_Y, R_Z, \dots$  and  $X, Y, Z, \dots$  and goals for  $\mathcal{V}$  we select the appropriate mix of processes  $p_i, q_j, r_k, \dots$  to satisfy the goals for  $\mathcal{V}$ , yielding a tailored

Process ( $\mathcal{P}\mathcal{V}$ ) ---> Product ( $\mathcal{V}$ )

The Quality Improvement Paradigm consists of six steps:

**Characterize** the current project and its environment with respect to models and metrics.

**Set** the quantifiable **goals** for successful project performance and improvement.

**Choose** the appropriate **process** model and supporting methods and tools for this project.

**Execute** the processes, construct the products, collect and validate the prescribed data, and analyze it to provide real-time feedback for corrective action.

**Analyze** the data to evaluate the current practices, determine problems, record findings, and make recommendations for future project improvements.

**Package** the experience in the form of updated and refined models and other forms of structured knowledge gained from this and prior projects and save it in an experience base to be reused on future projects.

The six steps of the Quality Improvement Paradigm can be combined in various ways to provide different views into the activities. First note that there are two feedback loops, a project feedback loop that takes place in the execution phase and an organizational feedback loop that takes place after a project is completed and changes the organization's understanding of the world between the packaging of what was learned from the last project and the characterization and baselining of the environment for the new project.

One high level organizational view of the paradigm is that we must **understand** (Characterize), **assess** (Set goals, Choose processes, Execute processes, Analyze data) and **package** (Package experience). Another view is to **plan** for a project (Characterize, Set goals, Choose processes), **develop** it (Execute processes), and then **learn** from the experience (Execute processes, Analyze data).

### **The Experience Factory Organization**

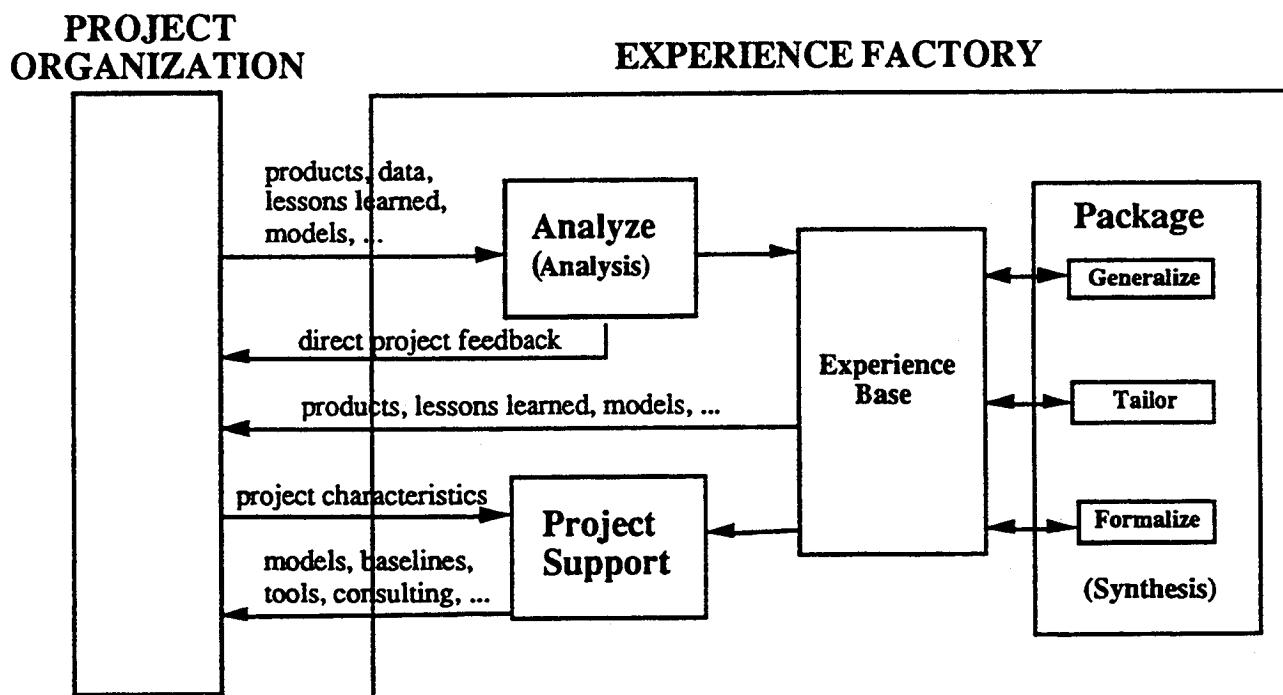
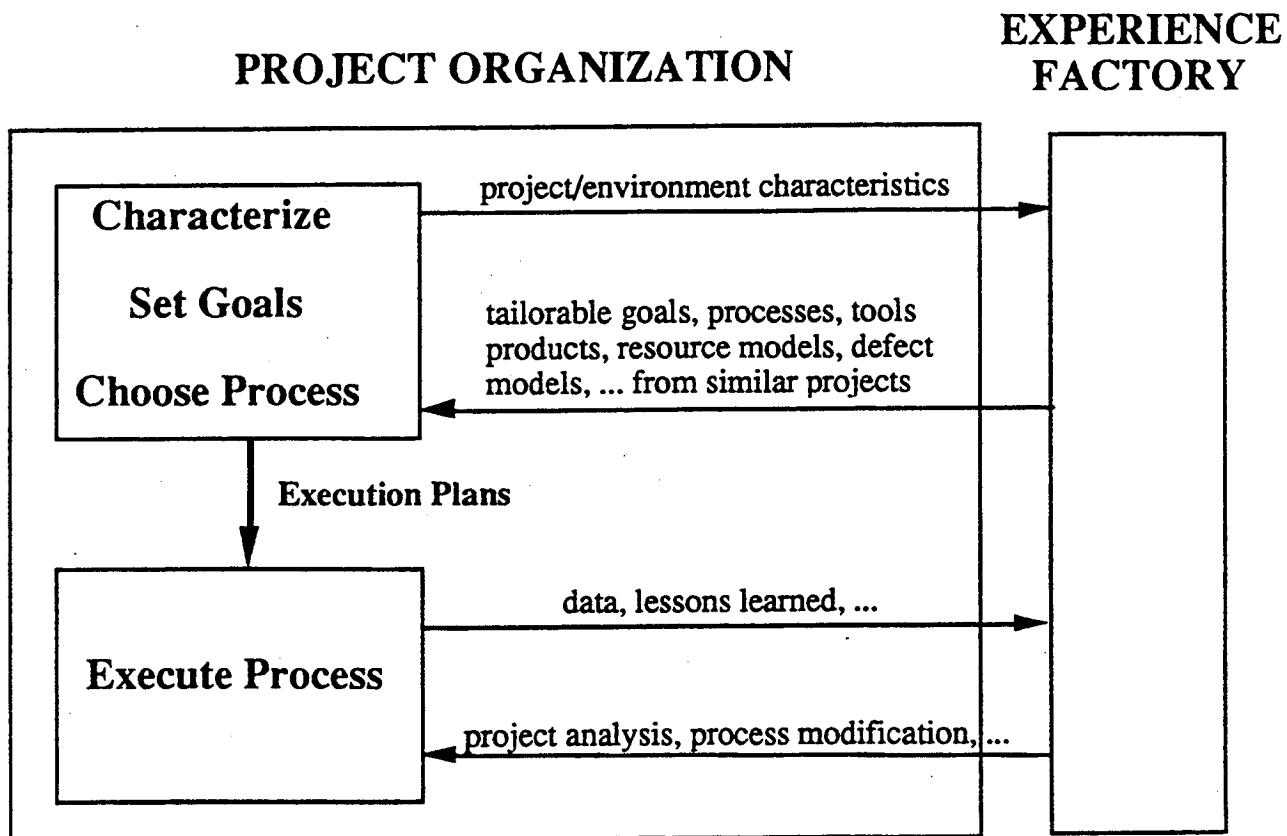
Note that the project personnel are primarily responsible for the planning and development activities (**Project Organization**) and a separate organization (**Experience Factory**) is primarily responsible for the learning and technology transfer activities. This provides the basis for the Experience Factory Organization. It recognizes the fact that the Quality Improvement Paradigm is really a paradigm shift for software development and requires a separate organization, i.e., an experience factory, whose job it is to package experience as opposed to problem solving - the job of the project organization. Problem solving involves such activities as the decomposition of a problem into simpler ones, instantiation, the design/implementation process, and validation and verification. Experience packaging involves such activities as the unification of different solutions and re-definition of the problem, generalization and formalization, the analysis/synthesis process, and experimentation.

It recognizes the fact that improving the software process and product requires the continual accumulation of evaluated experiences (learning), in a form that can be effectively understood and modified (experience models), stored in a repository of integrated experience models (experience base), that can be accessed/modified to meet the needs of the current project (reuse)

This systematic learning requires support for recording, *off-line* generalizing, tailoring, formalizing and synthesizing experience. Packaging and modeling useful experience requires a variety of models and formal notations that are tailorable, extendible, understandable, flexible and accessible. An effective experience base must contain accessible and integrated set of models that capture the *local* experiences. Systematic reuse requires support for using existing experience and on-line generalizing or tailoring of candidate experience.

This combination of ingredients requires an organizational structure that supports: a software

# EXPERIENCE FACTORY ORGANIZATION



evolution model that supports reuse processes for learning, packaging, and storing experience, and the integration of these two functions. It requires separate logical or physical organizations with different focuses/priorities, process models, expertise requirements: a Project Organization whose focus/priority is delivery supported by packaged reusable experiences, and an Experience Factory whose focus is to support project developments by analyzing and synthesizing all kinds of experience, acting as a repository for such experience, and supplying that experience to various projects on demand.

The Experience Factory packages experience by building informal, formal or schematized, and automated models and measures of various software processes, products, and other forms of knowledge via people, documents, and automated support.

### **Can the Quality Improvement Paradigm/Experience Factory Organization make you a level 5?**

How does the Quality Improvement Paradigm / Experience Factory Organization approach work in practice? You begin by putting the organization in place. This means collecting data to establish baselines, e.g., defects and resources, that are process and product independent and measuring your strengths and weaknesses to provide a business focus and goals for improvement, and establish product quality baselines. Using this information about your business, you select and experiment with methods and techniques to improve your processes based upon your product quality needs and evaluate your improvement based upon existing resource and defect baselines. You can define and tailor better and measurable processes, based upon the experience and knowledge gained within your own environment. You must measure for process conformance and domain understanding to make sure that your results are valid.

Now you will begin to understand the relationship between some process characteristics and product qualities and be able to manipulate some processes to achieve those product characteristics. As you change your processes you will establish new baselines and learn where the next place for improvement might be.

Thus, using the Quality Improvement Paradigm / Experience Factory Organization approach, you pull yourself up from the top rather than pushing up from the bottom. At step 1 you start with a level 5 style organization even though you do not yet have level 5 process capabilities. That is, you are driven by an understanding of your business, your product and process problems, your business goals, your experience with methods, etc. You learn from your business, not from an external model of process. You make process improvements based upon an understanding of the relationship between process and product in your organization. Technology infusion is motivated by the local problems, so people are more willing to try something new.

But what does a level 5 organization really mean? It is an organization that can manipulate process to achieve various product characteristics. This requires that we have a process and an organizational structure to help us:

- Understand our processes and products
- Measure and model the project and the organization
- Define and tailor process and product qualities explicitly
- Understand the relationship between process and product qualities
- Feedback information for project control
- Experiment with methods and techniques
- Evaluate our successes and failures
- Learn from our experiences

Package successful experiences  
Reuse successful experiences

So, can the Quality Improvement Paradigm/Experience Factory Organization make you a 5 in the Capability Maturity Model? Unfortunately, the answer is only *maybe*. You may not get a level 5 rating (depending on how a level 5 is defined when you get there - the definition is clearly evolving over time since there aren't many level 5 organizations at present) because your technologies are not from the "key set of processes" but you are operating at a level 5 definition and have chosen and tailored your processes to create a lean optimizing, continuously improving organization.

So what do you do if you want to be a level 5? Clearly, you can still use key process assessments to evaluate where you stand (along with your internal goals, needs, etc.). However, using the QIP/EF, the chances are your will move up the maturity scale faster. You will have more experience early on operating within an improvement organization structure, and you can demonstrate product improvement benefits early.

### Comparison of the frameworks

The Quality Improvement Paradigm / Experience Factory Organization can be compared with the other frameworks from a variety of points of view. First, it is similar to the Plan-Do-Check-Act paradigm in that it is evolutionary, based upon feedback loops, and learns from experiments. It is different in the sense that the Plan-Do-Check-Act paradigm is based upon production, i.e., it attempts to optimize a single process model/production line. In development, we rarely replicate the same thing twice. In production, we can collect a sufficient set of data based upon continual repetition of the same process to develop quantitative models of the process that will allow us to evaluate and predict quite accurately the effects of the single process model. We can use statistical quality control approaches. This is not possible for development, i.e. we must learn from one process about another, so our models are less rigorous and more abstract. Development processes are also more human based. This again effects the building, use, and accuracy of the types of models we can build.

The QIP/EF approach is compatible with TQM in that it can cover goals that are customer satisfaction driven and it is based upon the philosophy that quality is everyone's job. That is, everyone is part of the technology infusion process. Someone can be on the project team on one project and experimenting team on another. All the project personnel play the major role in the feedback mechanism. If they are not using the technology right it can be because they don't understand it, e.g., it wasn't taught right, it doesn't fit/interface with other project activities, it needs to be tailored, or it simply doesn't work. You need the user to tell you how to change it. This is consistent with the philosophy that no method is "packaged" that hasn't been tried (applied, analyzed, tailored).

The QIP/EF approach is most similar to the concepts of Lean Software Development in that it is based upon the ideas of tailoring a set of processes to meet particular problem/product under development. The goal is to generate an optimum set of processes, based upon models of the business and our experience about the relationship between process characteristics and product characteristics.

In summary, the QIP/EF approach provides for a separation of concerns/focus in differentiating between problem solving (the Project Organization) and experience modeling/packaging (the Experience Factory). It offers a support for learning and reuse and a means of formalizing and

integrating management and development technologies. It allows for the generation of a **tangible corporate asset**: an experience base of software competencies. It offers a Lean Software Development approach compatible with TQM while providing a level 5 CMM organizational structure. It links focused research with development. Best of all you can start small, evolve and expand, e.g., focus on a homogeneous set of projects or a particular set of packages and build from there.

## References

[Ba85a]

V. R. Basili, "Quantitative Evaluation of Software Engineering Methodology," Proc. of the First Pan Pacific Computer Conference, Melbourne, Australia, September 1985 [also available as Technical Report, TR-1519, Dept. of Computer Science, University of Maryland, College Park, July 1985].

[Ba89]

V. R. Basili, "Software Development: A Paradigm for the Future", Proceedings, 13th Annual International Computer Software & Applications Conference (COMPSAC), Keynote Address, Orlando, FL, September 1989

[BaRo87]

V. R. Basili, H. D. Rombach, "Tailoring the Software Process to Project Goals and Environments," Proc. of the Ninth International Conference on Software Engineering, Monterey, CA, March 30 - April 2, 1987, pp. 345-357.

[BaRo88]

V. R. Basili, H. D. Rombach "The TAME Project: Towards Improvement-Oriented Software Environments," IEEE Transactions on Software Engineering, vol. SE-14, no. 6, June 1988, pp. 758-773.

[BaCaMc92]

V. R. Basili, G. Caldiera, F. McGarry, R. Pajerski, G. Page, S. Waligora, "The Software Engineering Laboratory - an Operational Software Experience Factory", International Conference on Software Engineering, May, 1992, pp. 370-381.

[Cr80]

Philip B. Crosby, "Quality is free : the Art of Making Quality Certain," New American Library, New York, 1980.

[De86]

W. Edwards Deming, "Out of the Crisis," MIT Center for Advanced Engineering Study, MIT Press, Cambridge MA, 1986.

[Fe90]

Armand V. Feigenbaum, "Total Quality Control", fortieth anniversary edition, McGraw Hill, NY, 1991.

[Hu89]

Humphrey, Watts S., "Managing the software process," SEI series in software engineering, Addison-Wesley, Reading, Mass., 1989.

[Li67]

Rensis Likert, "The human organization: its Management and Value," McGraw Hill, New York, 1967.

[Sh31]

Walter A. Shewhart, "Economic control of quality of manufactured product," D. Van Nostrand Company, Inc., New York, 1931.

[RaHaMuPh85]

Ron Radice, A. J. Harding, P. E. Munnis, and R. W. Phillips, "A programming process study," IBM Systems Journal, vol.24, no. 2, 1985.

[WoJoRo90]

James P. Womack, Daniel T. Jones, Daniel Roos, "The machine that changed the world : based on the Massachusetts Institute of Technology 5-million dollar 5-year study on the future of the automobile," Rawson Associates, New York, 1990.

# The Experience Factory:

## Can it make you a 5?

or

What is its relationship to other Quality and  
Improvement Concepts?

Victor R. Basili

Institute for Advanced Computer Studies  
Department of Computer Science  
University of Maryland

© Victor R. Basili, 1992

## THE SOFTWARE BUSINESS

### Business Requirements

A successful business is a combination of technical and managerial solutions

Understand process and product  
*know the business*

Define process and product qualities  
*define the business needs to achieve them*

Feedback information for project control  
*define closed loop processes*

Evaluate successes and failures  
*evaluate every aspect of the business*

Learn from our experiences  
*each project should provide information that allows us to do business better*

Package successful experiences  
*build competencies in our areas of business*

Reuse successful experiences  
*use our competencies as the way we do business*

# THE SOFTWARE BUSINESS

---

## The Nature of Software

The software discipline is evolutionary and experimental; it is a laboratory science

Software technologies are human based

Software is development not production

There is a lack of models that allow us to reason about the process and the product

All software is not the same; process is a variable, goals are variable, ...

Packaged, reusable, experiences require a additional resources in the form of organization, processes, people, etc.

*Software is different; Software is difficult*

## ORGANIZATIONAL FRAMEWORKS

---

### Improving the Business

#### Plan-Do-Check-Act

a quality improvement process based upon a feedback cycle for optimizing a single process model/production line

#### Quality Improvement Paradigm/Experience Factory

continuous improvement through the experimentation, packaging and reuse of experiences based upon a business's needs

#### Total Quality Management

a management approach to long term success through customer satisfaction based on the participation of all members of an organization

#### SEI Capability Maturity Model

staged process improvement based upon assessment with regard to a set of key process areas until you reach a level 5 which represents a continuous process improvement

#### Lean (Software) Development

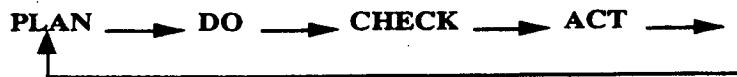
a principle supporting the concentration of the production on "value added" activities and the elimination or reduction of "not value added" activities

# ORGANIZATIONAL FRAMEWORKS

## Plan-Do-Check-Act Cycle (PDCA)

**Goal:** optimize/improve a single process model/production line based upon product feedback

**Approach:** uses such techniques as feedback loops and statistical quality control to experiment with methods for improvement and build predictive models of the product.



If Process ( $\mathcal{P}$ )  $\rightarrow$  Product ( $X$ ) then the approach yields

$$\mathcal{P}_0 \mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n \xrightarrow{\text{---}} X_0 X_1, X_2, \dots, X_n$$

where  $\mathcal{P}_i$  represents an improvement over  $\mathcal{P}_{i-1}$  and  $X_i$  has better quality than  $X_{i-1}$

**Notes:** Based upon work by W. A. Shewhart and made popular by W. E. Deming

# ORGANIZATIONAL FRAMEWORKS

## SEI Capability Maturity Model

**Goal:** to achieve a level 5 maturity rating, i.e., continuous process improvement via defect prevention, technology innovation, and process change management

**Approach:** A 5 level process maturity model is defined. Maturity level is defined based on repeated assessment of an organization's capability in key process areas. Improvement is achieved by action plans for poorly assessed processes.

Level	Focus
5 Optimizing	Continuous Process Improvement
4 Managed	Product & Process Quality
3 Defined	Engineering Process
2 Repeatable	Project Management
1 Initial	Heres

If Process ( $\mathcal{P}$ ) is level  $i$  then modify the process based upon the the key processes of the model until the porcess model is at level  $i+1$ .

**Notes:** Organizational and quality management maturity models were developed by R. Likert and P. Crosby, respectively. A software maturity model was developed by Ron Radice while he was at IBM. It was made popular by Watts Humphrey at the SEI.

# ORGANIZATIONAL FRAMEWORKS

---

## Lean Software Development

**Goal:** to build software using the minimal set of activities needed, eliminating non essential steps, i.e., tailoring the process to the product needs

**Approach:** uses such concepts as technology management, human centered management, decentral organization, quality management, supplier and customer integration, and internationalization/regionalization.

Given the characteristics for product  $\mathcal{V}$

select the appropriate mix of subprocesses  $p_i, q_j, r_k, \dots$  to satisfy the goals for  $\mathcal{V}$ ,

yielding a minimal tailored Process ( $\mathcal{P}_V$ )  $\xrightarrow{\text{-----}}$  Product ( $\mathcal{V}$ )

**Notes:** Based upon Lean Enterprise Management, a philosophy that has been used to improve factory output. Womack, et. al. (1989), have written a book on the application of lean enterprises in the automotive industry.

# ORGANIZATIONAL FRAMEWORKS

---

## Total Quality Management

**Goal:** generate institutional commitment to success through customer satisfaction

**Approach:** varied, Hughes\* uses such techniques as Quality Function Deployment (QFD), design of experiments (DOE), and statistical process control (SPC), to improve the product through the process.



If Process ( $\mathcal{P}$ )  $\rightarrow$  Product ( $\mathcal{X}$ ) then the approach yields

$\mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n \xrightarrow{\text{-----}} \mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n$

where  $\mathcal{P}_i$  represents an improvement over  $\mathcal{P}_{i-1}$  and  $\mathcal{X}_i$  provides better customer satisfaction than  $\mathcal{X}_{i-1}$

**Notes:** The term TQM was coined by the Naval Air Systems Command in 1985 to describe its Japanese style management approach to quality improvement.

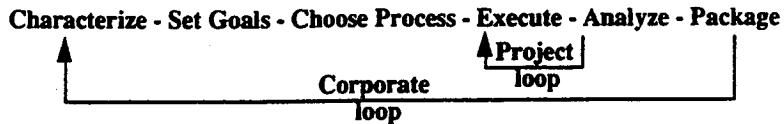
\*Source: Michael Deutsch

# ORGANIZATIONAL FRAMEWORKS

## Quality Improvement Paradigm

**Goal:** build a continually improving organization based upon its evolving goals and an assessment of its status relative to those goals

**Approach:** uses internal assessment against the organization's own goals and status (rather than process areas) and such techniques as GQM, model building, qualitative/quantitative analysis to improve the product through the process

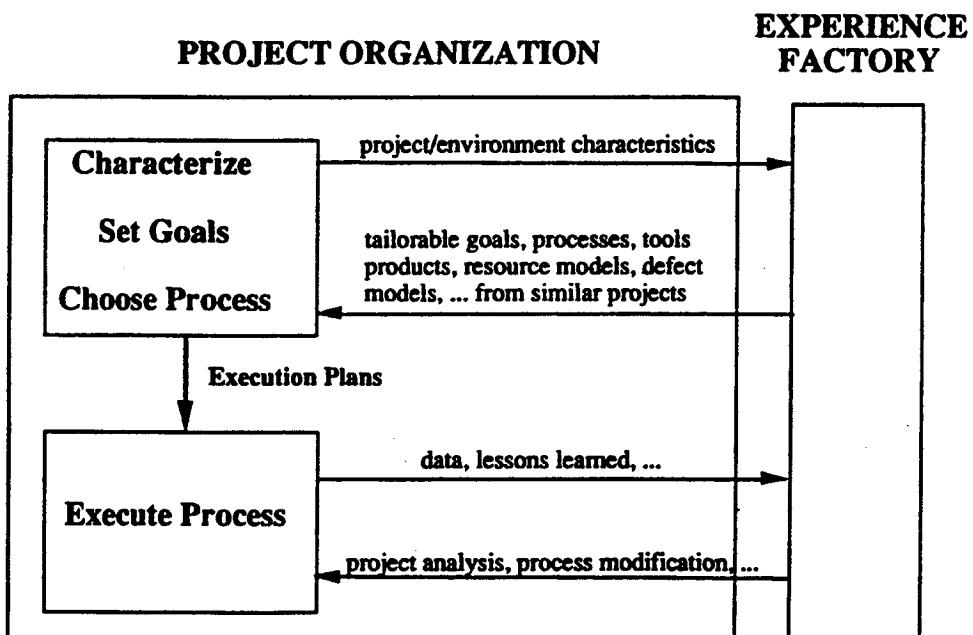


If Processes  $(P_X, Q_Y, R_Z, \dots)$   $\rightarrow$  Products  $(X, Y, Z, \dots)$  and we want to build  $V$  then based upon an understanding of the relationship between  $P_X, Q_Y, R_Z, \dots$  and  $X, Y, Z, \dots$  and goals for  $V$  we select the appropriate mix of processes  $p_i, q_j, r_k, \dots$  to satisfy the goals for  $V$ , yielding a tailored

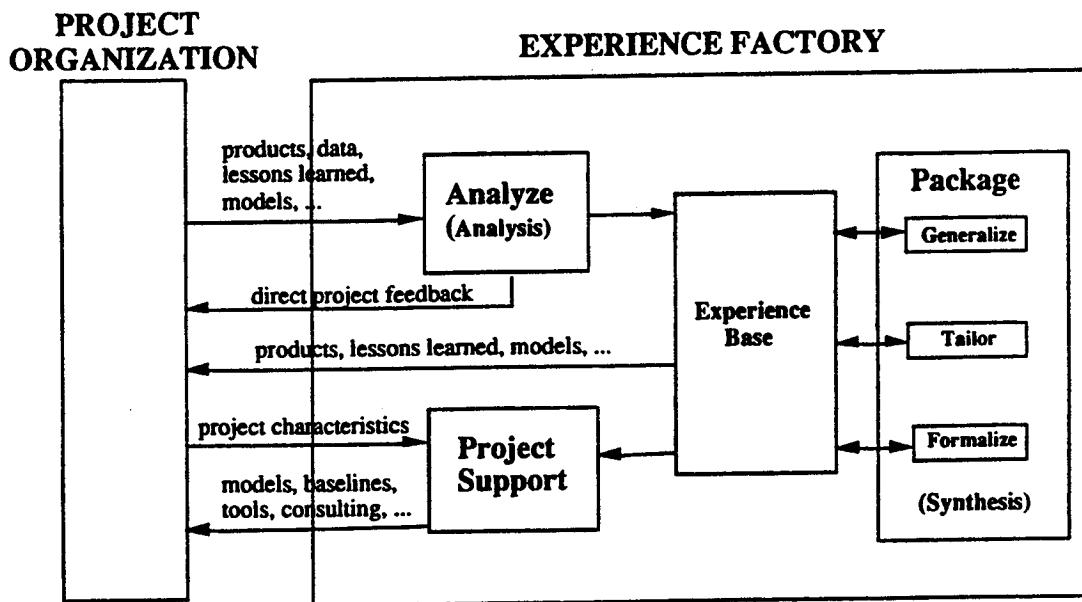
Process  $(P_V)$   $\rightarrow$  Product  $(V)$

Note: The approach has evolved over 17 years based upon lessons learned in the SEL.

## EXPERIENCE FACTORY ORGANIZATION



# EXPERIENCE FACTORY ORGANIZATION



# EXPERIENCE FACTORY ORGANIZATION

## A Different Paradigm

### Project Organization Problem Solving

Decomposition of a problem  
into simpler ones

Instantiation

Design/Implementation process

Validation and Verification

### Experience Factory Experience Packaging

Unification of different solutions  
and re-definition of the problem

Generalization, Formalization

Analysis/Synthesis process

Experimentation

# **EXPERIENCE FACTORY ORGANIZATION**

---

## **What do you do?**

Put the organization in place, collect data to establish baselines, e.g., defects and resources, that are process and product independent.

Measure your strengths and weaknesses, provide a business focus and goals for improvement, and establish product quality baselines.

Select and experiment with methods and techniques to improve process based upon product quality needs and evaluate improvement based upon existing resource and defect baselines.

Define and tailor better and measurable processes, based upon experience and knowledge of the environment, process conformance and domain understanding.

Now you will begin to understand the relationship between some process characteristics and product qualities and be able to manipulate some processes to achieve those product characteristics.

As you change your processes you will establish new baselines and learn where the next place for improvement might be.

## **ORGANIZATIONAL FRAMEWORKS**

---

### **Comparison based on the nature of software**

Which frameworks assume an evolutionary and experimental discipline?

Are the paradigm assumptions explicitly based upon the more general idea of software development rather than production?

Does the paradigm explicitly recognize that the technologies are human based?

Does the paradigm explicitly help in the development of models and abstractions of the discipline? Does the paradigm explicitly assume that packaging reusable experiences requires a separate organization and resources?

Does the paradigm explicitly support learning across the differences in software projects, goals, organizations,etc.?

# **EXPERIENCE FACTORY ORGANIZATION**

---

## **Can it make you a 5?**

**Using the Quality Improvement Paradigm / Experience Factory Organization:**

**You pull yourself up from the top rather than pushing up from the bottom**

**At step 1 you start with a level 5 organization but not level 5 capabilities**

**You are driven by an understanding of your business, your product and process problems, your business goals, your experience with methods, etc.**

**You learn from your business, not on an external model of process**

**You make process improvements based upon an understanding of the relationship between process and product in your organization**

# **EXPERIENCE FACTORY ORGANIZATION**

---

## **Can it make you a 5?**

**What does a level 5 organization mean?**

**It is an organization that can manipulate process to achieve various product characteristics.**

**This requires that we have a process and an organizational structure to help us:**

**Understand our processes and products**

**Measure and model the project and the organization**

**Define and tailor process and product qualities explicitly**

**Understand the relationship between process and product qualities**

**Feedback information for project control**

**Experiment with methods and techniques**

**Evaluate our successes and failures**

**Learn from our experiences**

**Package successful experiences**

**Reuse successful experiences**

## Can it make you a 5?

You may not get a level 5 rating (depending on how it gets defined when you get there) because your technologies are not from the "key set of processes" but you are operating at a level 5 definition and have chosen and tailored your processes to create a lean optimizing, continuously improving organizations.

## How does this fit in with the CMM?

You can still use key process assessments to evaluate where you stand (along with your internal goals, needs, etc.) and chances are you will move up the maturity scale faster

You will have more experience early on with an improvement organization, and you can demonstrate product improvement benefits early